ELSEVIER
Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®
Theoretical Computer Science 304 (2003) 237–247
Theoretical Computer Science
www.elsevier.com/locate/tcs

A PTAS for weight constrained Steiner trees in series–parallel graphs

Guangting Chen[a,1], Guoliang Xue[b,*,2]

[a] *School of Science and Arts, Hangzhou Institute of Electronic Engineering, Hangzhou, China*
[b] *Department of Computer Science and Engineering, Arizona State University, P.O. Box 875406, Tempe, AZ 85287-5406, USA*

Received 9 January 2002; accepted 14 December 2002
Communicated by D.-Z. Du

Abstract

In this paper, we study the problem of computing a minimum cost Steiner tree subject to weight constraint in a series–parallel graph where each edge has a nonnegative integer cost and a nonnegative integer weight. We present a strongly polynomial time approximation scheme for this NP-complete problem.

Keywords: Multiple constraints quality of service; Steiner trees; Series–parallel graphs

1. Introduction

A computer network is often modeled by an undirected graph $G=(V,E)$, where V is the set of vertices and E is the set of edges. The traditional Steiner tree problem associates an edge cost $c(e)\geqslant 0$ with each edge $e\in E$ and asks for a minimum cost subgraph of G spanning a given subset $\mathcal{T}\subseteq V$ of *target* vertices. Such problems find important applications in computer networks and have been studied by many researchers [1,4,10,14,16]. We refer the readers to [8,11] for details.

* Corresponding author.
E-mail addresses: gtchen@mail.hz.zj.cn (Guangting Chen), xue@cse.asu.edu (Guoliang Xue).
[1] Supported in part by Chinese NSF grant CNSF19971078 and Chinese Institute of Electronic Science grant 57-7-3-7.
[2] Supported in part by ARO grant DAAD19-00-1-0377 and by DOE grant DE-FG02-00ER45828. A preliminary version of this paper was presented at COCOON'01 where we presented a PTAS whose running time also depends on the input size of the problem.

0304-3975/03/$ - see front matter
doi:10.1016/S0304-3975(03)00088-4

In this traditional network model, there is only one cost corresponding to each edge. In real-life problems, each edge may have a weight besides a cost, and we want to find a minimum cost subgraph of G spanning the vertices in $\mathscr{T}$ with a total weight no more than a given weight constraint $\mathscr{W}$. We call this problem the *weight constrained minimum cost Steiner tree problem* (WCSTP).

Formally, we generalize the traditional network model to allow two independent edge weighting functions: with each edge $e \in E$, there is an associated integer *cost* $c(e) \geqslant 0$ and an associated integer *weight* $w(e) \geqslant 0$. Let G' be a subgraph of G. The cost of G', denoted by $c(G')$, is the sum of the edge costs of G'. The weight of G', denoted by $w(G')$, is the sum of the edge weights of G'. Given a set of *target vertices* $\mathscr{T} \subseteq V$ and an integer weight constraint $\mathscr{W} \geqslant 0$, we are interested in computing a minimum cost tree subgraph T of G which spans the vertices in $\mathscr{T}$ subject to the constraint that $w(T) \leqslant \mathscr{W}$. We call such a tree a *weight constrained minimum cost Steiner minimum tree*.

When $w(e) = 0$ for every edge $e \in E$, the WCSTP becomes the well-known Steiner tree problem. Since the Steiner tree problem is NP-hard in the strong sense [6], the WCSTP is also NP-hard in strong sense. In this paper, we are interested in WCSTP on a very important class of sparse networks known as series–parallel graphs (S–P graphs), which are subgraphs of 2-trees defined below.

Following [14], a *2-tree* can be defined recursively as follows, and all 2-trees may be obtained in this way. A triangle (a complete graph on three vertices) is a 2-tree. Given a 2-tree and an edge $\{x, y\}$ of the 2-tree, we can add a new vertex z adjacent to both x and y; the result is a 2-tree. An S–P graph (also known as a *partial 2-tree*) is a subgraph of a 2-tree. With this definition, one can see that a 2-tree on n vertices has $2n - 3$ edges and $n - 2$ triangles.

Farley [5] demonstrated that 2-trees are isolated failure immune (IFI) networks. Wald and Colbourn [14] showed that a minimum IFI network is a 2-tree. This fact made 2-trees an important class of fault tolerant networks. Wald and Colbourn [14] also showed that the Steiner tree problem on an S–P graph is tractable, presenting a linear time algorithm. In a recent paper, Colbourn and Xue [3] showed that the *grade of service Steiner tree problem* [17] is also tractable on an S–P graph.

In this paper, we show that the WCSTP is NP-hard even when the graph G is an S–P graph. We then present a strongly polynomial time approximation scheme for computing a weight constrained minimum cost Steiner tree in an S–P graph.

2. WCSTP in S–P graphs is intractable

When the number of target vertices is zero or one, the WCSTP is trivial. When there are only two vertices in $\mathscr{T}$, the WCSTP becomes the well-known *weight constrained shortest path problem* (WCSPP) which is known to be NP-hard [6]. To the best of our knowledge, no one has ever addressed the complexity of WCSTP or WCSPP in an S–P graph specifically. In a recent paper [15], Wang and Crowcroft presented an NP-hardness proof for the WCSPP problem in general graphs. We point out that the NP-hardness proof of Wang and Crowcroft, *although targeted at the problem in* *general graphs*, also proves that the WCSPP is NP-hard in an S–P graph. Since the WCSPP in an S–P graph is NP-hard, the more general WCSTP in an S–P graph is also NP-hard. Therefore, we have the following hardness result (for the completeness of the current paper, we include an NP-hardness proof of the WCSPP in an S–P graph in the appendix).

Theorem 2.1. *The WCSTP in an S–P graph is NP-hard.*

Note that in the definition of the WCSTP, the weight and cost of an edge are symmetric. Therefore, we may also talk about *cost constrained minimum weight Steiner tree problem* (CCSTP) which asks for a minimum weight tree subject to cost constraint. Given the hardness of the problem, we are interested in designing efficient approximation algorithms for this problem. In the next section, we will present a pseudo-polynomial time algorithm for computing a cost constrained minimum weight Steiner tree in an S–P graph. We will then apply standard techniques of scaling and rounding to turn the pseudo-polynomial time algorithm into a strongly polynomial time approximation scheme (PTAS) for weight constrained minimum cost Steiner trees.

3. A pseudo-polynomial time algorithm for CCSTP

In this section, we will present a pseudo-polynomial time algorithm for computing a cost constrained minimum weight Steiner tree. Given the set $\mathscr{T} \subseteq V$ and a nonnegative integer ζ, the algorithm computes a minimum weight Steiner tree among those whose cost is bounded by ζ, in $O(n(\zeta+1)^2)$ time, where n is the number of vertices in the series–parallel graph.

Our algorithm for computing a cost constrained Steiner tree operates in two phases. The first completes the graph to a 2-tree, as was done in [14], where each added edge has a weight of zero and a cost that is larger than the cost constraint ζ. This ensures that the added edges will never be chosen in a cost constrained minimum weight Steiner tree. The second phase involves finding cost constrained minimum weight Steiner trees in 2-trees.

We are given a 2-tree G with target set $\mathscr{T} \subseteq V$. With each (directed) arc $\alpha = (x, y)$ corresponding to an (undirected) edge $\{x, y\}$ of G, we will associate $6\zeta + 6$ measures, which summarize the weight incurred so far in the subgraph S which has been reduced onto the edge $\{x, y\}$:

- $st(\alpha, \xi)$ is the minimum weight of a Steiner tree for S, with a cost of at most ξ, in which x and y appear in the same tree, $\xi = 0, 1, 2, \ldots, \zeta$;
- $dt(\alpha, \xi)$ is the minimum weight of two disjoint trees for S including all targets, one tree involving x and the other involving y, whose total cost is no more than ξ, $\xi = 0, 1, 2, \ldots, \zeta$;
- $yn(\alpha, \xi)$ is the minimum weight of a Steiner tree for S, which includes x but not y, whose cost is at most ξ, $\xi = 0, 1, 2, \ldots, \zeta$;
- $ny(\alpha, \xi)$ is the minimum weight of a Steiner tree for S, which includes y but not x, whose cost is at most ξ, $\xi = 0, 1, 2, \ldots, \zeta$;

- $nn(\alpha,\xi)$ is the minimum weight of a Steiner tree for S, which includes neither x nor y, whose cost is at most ξ, $\xi=0,1,2,\ldots,\zeta$;
- $none(\alpha,\xi)$ is the weight of omitting all vertices of S from a Steiner tree whose cost is at most ξ, $\xi=0,1,2,\ldots,\zeta$ (note that there is a big penalty here if S contains any target vertices).

Let $\beta=(x,y)$ be the other arc corresponding to the edge $\{x,y\}$. Then the $6\zeta+6$ measures associated with β relates to the measures associated with α in the following way:

$st(\beta,\xi)=st(\alpha,\xi)$	$dt(\beta,\xi)=dt(\alpha,\xi)$	$yn(\beta,\xi)=ny(\alpha,\xi)$
$ny(\beta,\xi)=yn(\alpha,\xi)$	$nn(\beta,\xi)=nn(\alpha,\xi)$	$none(\beta,\xi)=none(\alpha,\xi)$

Let BIG be an integer which is bigger than the sum of all the edge weights in the given graph. Initially, no reduction of the graph has been done. We set the initial measures for each arc $\alpha=(x,y)$ corresponding to an edge $e=\{x,y\}$ as follows, for $\xi=0,1,2,\ldots,\zeta$.

Our algorithm (call it **Algorithm 1**) will use these initial arc measures to recompute the arc measures as the graph is reduced. The graph is reduced by repeated deletion of degree-2 vertices. The algorithm stops when there is no degree-2 vertex left. Suppose at some point in the algorithm there is a triangle $\{x,y,z\}$ in which y is a degree-2 vertex and that costs have been computed for both (x,y) and (y,z). Let $L=(x,y)$, $R=(y,z)$, $M=(x,z)$. We will update the measures at M using the following rules:

- For $\xi=0,1,2,\ldots,\zeta$, let $w_{\xi,st}$ be the minimum of the following four sets of numbers:

$$\{st(M,\xi_M)+st(L,\xi_L)+dt(R,\xi_R)|\xi_L+\xi_R+\xi_M=\xi,\ \xi_L\geqslant 0,\ \xi_R\geqslant 0,\ \xi_M\geqslant 0\};$$
$$\{st(M,\xi_M)+dt(L,\xi_L)+st(R,\xi_R)|\xi_L+\xi_R+\xi_M=\xi,\ \xi_L\geqslant 0,\ \xi_R\geqslant 0,\ \xi_M\geqslant 0\};$$
$$\{st(M,\xi_M)+yn(L,\xi_L)+ny(R,\xi_R)|\xi_L+\xi_R+\xi_M=\xi,\ \xi_L\geqslant 0,\ \xi_R\geqslant 0,\ \xi_M\geqslant 0\};$$
$$\{dt(M,\xi_M)+st(L,\xi_L)+st(R,\xi_R)|\xi_L+\xi_R+\xi_M=\xi,\ \xi_L\geqslant 0,\ \xi_R\geqslant 0,\xi_M\geqslant 0\}.$$

- For $\xi=0,1,2,\ldots,\zeta$, let $w_{\xi,dt}$ be the minimum of the following three sets of numbers:

$$\{dt(M,\xi_M)+st(L,\xi_L)+dt(R,\xi_R)|\xi_L+\xi_R+\xi_M=\xi,\ \xi_L\geqslant 0,\ \xi_R\geqslant 0,\ \xi_M\geqslant 0\};$$
$$\{dt(M,\xi_M)+dt(L,\xi_L)+st(R,\xi_R)|\xi_L+\xi_R+\xi_M=\xi,\ \xi_L\geqslant 0,\ \xi_R\geqslant 0,\ \xi_M\geqslant 0\};$$
$$\{dt(M,\xi_M)+yn(L,\xi_L)+ny(R,\xi_R)|\xi_L+\xi_R+\xi_M=\xi,\ \xi_L\geqslant 0,\ \xi_R\geqslant 0,\ \xi_M\geqslant 0\}.$$

- For $\xi = 0,1,2,\ldots,\zeta$, let $w_{\xi,yn}$ be the minimum of the following two sets of numbers:

$$\{yn(M,\xi_M) + yn(L,\xi_L) + none(R,\xi_R) | \xi_L + \xi_R + \xi_M = \xi,\\ \xi_L \geqslant 0,\ \xi_R \geqslant 0,\ \xi_M \geqslant 0\};$$
$$\{yn(M,\xi_M) + st(L,\xi_L) + yn(R,\xi_R) | \xi_L + \xi_R + \xi_M = \xi,\\ \xi_L \geqslant 0,\ \xi_R \geqslant 0,\ \xi_M \geqslant 0\}.$$

- For $\xi = 0,1,2,\ldots,\zeta$, let $w_{\xi,ny}$ be the minimum of the following two sets of numbers:

$$\{ny(M,\xi_M) + ny(L,\xi_L) + st(R,\xi_R) | \xi_L + \xi_R + \xi_M = \xi,\\ \xi_L \geqslant 0,\ \xi_R \geqslant 0,\ \xi_M \geqslant 0\};$$
$$\{ny(M,\xi_M) + none(L,\xi_L) + ny(R,\xi_R) | \xi_L + \xi_R + \xi_M = \xi,\\ \xi_L \geqslant 0,\ \xi_R \geqslant 0,\ \xi_M \geqslant 0\}.$$

- For $\xi = 0,1,2,\ldots,\zeta$, let $w_{\xi,nn}$ be the minimum of the following four sets of numbers:

$$\{nn(M,\xi_M) + none(L,\xi_L) + none(R,\xi_R) | \xi_L + \xi_R + \xi_M = \xi,\\ \xi_L \geqslant 0,\ \xi_R \geqslant 0,\ \xi_M \geqslant 0\};$$
$$\{none(M,\xi_M) + nn(L,\xi_L) + none(R,\xi_R) | \xi_L + \xi_R + \xi_M = \xi,\\ \xi_L \geqslant 0,\ \xi_R \geqslant 0,\ \xi_M \geqslant 0\};$$
$$\{none(M,\xi_M) + none(L,\xi_L) + nn(R,\xi_R) | \xi_L + \xi_R + \xi_M = \xi,\\ \xi_L \geqslant 0,\ \xi_R \geqslant 0,\ \xi_M \geqslant 0\};$$
$$\{none(M,\xi_M) + ny(L,\xi_L) + yn(R,\xi_R) | \xi_L + \xi_R + \xi_M = \xi,\\ \xi_L \geqslant 0,\ \xi_R \geqslant 0,\ \xi_M \geqslant 0\}.$$

- For $\xi = 0,1,2,\ldots,\zeta$, let $w_{\xi,none}$ be the minimum of the following set of numbers:

$$\{none(M,\xi_M) + none(L,\xi_L) + none(R,\xi_R) | \xi_L + \xi_R + \xi_M = \xi,\\ \xi_L \geqslant 0,\ \xi_R \geqslant 0,\ \xi_M \geqslant 0\}.$$

- Delete the vertex y from the graph. For $\xi = 0,1,2,\ldots,\zeta$, update the values of $st(M,\xi)$, $dt(M,\xi)$, $yn(M,\xi)$, $ny(M,\xi)$, $nn(M,\xi)$, and $none(M,\xi)$ to $w_{\xi,st}$, $w_{\xi,dt}$, $w_{\xi,yn}$, $w_{\xi,ny}$, $w_{\xi,nn}$, and $w_{\xi,none}$, respectively.

Theorem 3.1. *When the* 2*-tree is reduced to a single edge* $\alpha = \{x,y\}$*, the minimum of the four values* $st(\alpha,\zeta)$*,* $yn(\alpha,\zeta)$*,* $ny(\alpha,\zeta)$*,* $nn(\alpha,\zeta)$ *is the minimum weight of a Steiner tree interconnecting all the vertices in* $\mathscr{T}$ *with a cost not more than* ζ*, where a weight of* BIG *or more indicates the nonexistence of a cost constrained Steiner tree.*

Proof. We note that the measures of an edge $\alpha = \{x,y\}$ are initialized correctly, and are updated correctly after each reduction. For example, for any $\xi \in \{0,1,2,\ldots,\zeta\}$ and an edge $\alpha = \{x,y\}$, $st(\alpha,\xi)$ is the minimum weight of a Steiner tree for S whose cost is no more than ξ, in which x and y appear in the same tree.

When the graph is reduced to a single arc $\alpha = \{x, y\}$, $st(\alpha, \zeta)$ is the minimum weight of a cost constrained Steiner tree interconnecting x, y, and all the vertices in $\mathscr{T}$; $yn(\alpha, \zeta)$ is the minimum weight of a cost constrained Steiner tree interconnecting x (but not y) and all the vertices in $\mathscr{T} - \{y\}$; $ny(\alpha, \zeta)$ is the minimum weight of a cost constrained Steiner tree interconnecting y (but not x) and all the vertices in $\mathscr{T} - \{x\}$; $nn(\alpha, \zeta)$ is the minimum weight of a cost constrained Steiner tree interconnecting all the vertices in $\mathscr{T} - \{x, y\}$ (but not x nor y). Note that leaving out a target vertex from the tree receives a penalty of at least BIG. Therefore, the minimum of the above four values corresponds to the minimum weight of a cost constrained Steiner tree. □

Theorem 3.2. *Algorithm* 1 *can be implemented in* $O(n(\zeta + 1)^2)$ *time*, *where* ζ *is the cost constraint and n is the number of vertices in the graph.*

Proof. We will show that the update of measurements can be accomplished in $O((\zeta + 1)^2)$ time for each reduction. Since there are $O(n)$ reductions, this would complete the proof of the theorem.

Since there are $O(\zeta + 1)$ choices for each of ξ, ξ_M and ξ_L ($\xi_R = \xi - \xi_M - \xi_L$), a simple minded implementation would require $O((\zeta + 1)^3)$ time to update the measures for each reduction.

Let us denote $\xi_L + \xi_R$ by ξ_{LR}. Although there are $O((\zeta + 1)^2)$ different choices for (ξ_L, ξ_R), there are only $O(\zeta + 1)$ different choices for ξ_{LR}. Using $O((\zeta + 1)^2)$ time, we can compute the following $12(\zeta + 1)$ measures for all $\xi_{LR} \in \{0, 1, 2, \ldots, \zeta\}$:

$$LR(\xi_{LR}, st, st) = \min\{st(L, \xi_L) + st(R, \xi_{LR} - \xi_L) | 0 \leqslant \xi_L \leqslant \xi_{LR}\};$$
$$LR(\xi_{LR}, st, dt) = \min\{st(L, \xi_L) + dt(R, \xi_{LR} - \xi_L) | 0 \leqslant \xi_L \leqslant \xi_{LR}\};$$
$$LR(\xi_{LR}, dt, st) = \min\{dt(L, \xi_L) + st(R, \xi_{LR} - \xi_L) | 0 \leqslant \xi_L \leqslant \xi_{LR}\};$$
$$LR(\xi_{LR}, yn, ny) = \min\{yn(L, \xi_L) + ny(R, \xi_{LR} - \xi_L) | 0 \leqslant \xi_L \leqslant \xi_{LR}\};$$
$$LR(\xi_{LR}, yn, none) = \min\{yn(L, \xi_L) + none(R, \xi_{LR} - \xi_L) | 0 \leqslant \xi_L \leqslant \xi_{LR}\};$$
$$LR(\xi_{LR}, st, yn) = \min\{st(L, \xi_L) + yn(R, \xi_{LR} - \xi_L) | 0 \leqslant \xi_L \leqslant \xi_{LR}\};$$
$$LR(\xi_{LR}, ny, st) = \min\{ny(L, \xi_L) + st(R, \xi_{LR} - \xi_L) | 0 \leqslant \xi_L \leqslant \xi_{LR}\};$$
$$LR(\xi_{LR}, none, ny) = \min\{none(L, \xi_L) + ny(R, \xi_{LR} - \xi_L) | 0 \leqslant \xi_L \leqslant \xi_{LR}\};$$
$$LR(\xi_{LR}, none, none) = \min\{none(L, \xi_L) + none(R, \xi_{LR} - \xi_L) | 0 \leqslant \xi_L \leqslant \xi_{LR}\};$$
$$LR(\xi_{LR}, nn, none) = \min\{nn(L, \xi_L) + none(R, \xi_{LR} - \xi_L) | 0 \leqslant \xi_L \leqslant \xi_{LR}\};$$
$$LR(\xi_{LR}, none, nn) = \min\{none(L, \xi_L) + nn(R, \xi_{LR} - \xi_L) | 0 \leqslant \xi_L \leqslant \xi_{LR}\};$$
$$LR(\xi_{LR}, ny, yn) = \min\{ny(L, \xi_L) + yn(R, \xi_{LR} - \xi_L) | 0 \leqslant \xi_L \leqslant \xi_{LR}\}.$$

After those $12(\zeta + 1)$ measures are computed, only $O((\zeta + 1)^2)$ time are necessary to compute the updated values for $st(M, \xi)$, $dt(M, \xi)$, $yn(M, \xi)$, $ny(M, \xi)$, $nn(M, \xi)$, $none(M, \xi)$, for $\xi = 0, 1, 2, \ldots, \zeta$. For example, $w_{\xi, yn}$ can be computed as the minimum of

$$\min\{yn(M, \xi_M) + LR(\xi - \xi_M, yn, none) | 0 \leqslant \xi_M \leqslant \xi\}$$

and

$$\min\{yn(M,\xi_M)+LR(\xi-\xi_M,st,yn)|0\leqslant\xi_M\leqslant\xi\}.$$

This shows that Algorithm 1 can be implemented in $O(n(\zeta+1)^2)$ time. $\square$

We point out that the cost constrained Steiner tree can be constructed in $O(n(\zeta+1))$ extra time if we perform some bookkeeping operations (recording how the minimum was achieved) during the reductions. If we fix ζ at 0 and assume the cost of all edges in the graph are zero, the above algorithm finds a minimum weight Steiner tree in $O(n)$ time. This is why we choose to use $\zeta+1$ instead of ζ in our complexity analysis.

4. A PTAS for WCSTP

We use standard techniques of scaling and rounding [2,7,9,13,18] to turn the pseudo-polynomial time algorithm for CCSTP into a strongly polynomial time approximation scheme for WCSTP.

Let us use $c(\mathscr{T},\mathscr{W})$ to denote the minimum cost of a Steiner tree spanning the targets in $\mathscr{T}$ with a weight of no more than $\mathscr{W}$. Given a positive real number $\mathscr{C}$, deciding whether $c(\mathscr{T},\mathscr{W})>\mathscr{C}$ or $c(\mathscr{T},\mathscr{W})\leqslant\mathscr{C}$ is NP-hard. Using the standard technique of scaling and rounding [2,7,9,13,18], we can decide, in strongly polynomial time, whether $c(\mathscr{T},\mathscr{W})>\mathscr{C}$ or $c(\mathscr{T},\mathscr{W})<(1+\varepsilon)\mathscr{C}$, for any given constant $\varepsilon>0$. This technique will play an important role in our PTAS for computing a weight constrained minimum cost Steiner tree in an S–P graph. We describe this approximate testing in Algorithm 2 as TEST.

Algorithm 2. TEST($\mathscr{C},\varepsilon$)

Step_1 Set $\theta:=\frac{n}{\mathscr{C}\times\varepsilon}$; Let c_θ be the *scaled edge cost function* such that $c_\theta(e)=\lfloor c(e)\times\theta\rfloor$ for $e\in E$; Set $\zeta:=\mathscr{C}\times\theta$;

Step_2 Apply Algorithm 1 using the scaled edge cost function c_θ instead of the original edge cost function c;

```
if the weight of the cost constrained Steiner tree is no more than 𝒲 then
   output YES ;
else
   output NO ;
endif
```

Theorem 4.1. *Let us be given the target set $\mathscr{T}$, the weight constraint $\mathscr{W}$, the positive real numbers $\mathscr{C}$ and ε. If* TEST($\mathscr{C},\varepsilon$) = NO, *then $c(\mathscr{T},\mathscr{W})>\mathscr{C}$. If* TEST($\mathscr{C},\varepsilon$) = YES, *then $c(\mathscr{T},\mathscr{W})<(1+\varepsilon)\times\mathscr{C}$. In addition, the worst-case time complexity of* TEST($\mathscr{C},\varepsilon$) *is* $O(n^3/\varepsilon^2)$.

Proof. Let T be a tree subgraph in G. Let $c(T)=\sum_{e\in T}c(e)$ and $c_\theta(T)=\sum_{e\in T}c_\theta(e)$. Since, the number of edges in T is at most $n-1$, we can prove that $c_\theta(T)\leqslant\zeta$ implies $c(T)\leqslant\mathscr{C}(1+\varepsilon)$ and that $c_\theta(T)>\zeta$ implies $c(T)>\mathscr{C}$, where θ and ζ are as defined in Algorithm 2.

Assume that $\mathsf{TEST}(\mathscr{C},\varepsilon)=\mathsf{NO}$. Then we know that for any tree T spanning the targets in $\mathscr{T}$ with a weight no more than $\mathscr{W}$, we must have $c_\theta(T)>\zeta$, which in turn implies $c(T)>\mathscr{C}$. This says that $c(\mathscr{T},\mathscr{W})>\mathscr{C}$.

Now assume that $\mathsf{TEST}(\mathscr{C},\varepsilon)=\mathsf{YES}$. Then we know that there is a tree T spanning the targets in $\mathscr{T}$ such that $w(T)\leqslant\mathscr{W}$ and $c_\theta(T)\leqslant\zeta$. Note that $c_\theta(T)\leqslant\zeta$ implies $c(T)<\mathscr{C}(1+\varepsilon)$. This says that $c(\mathscr{T},\mathscr{W})<\mathscr{C}(1+\varepsilon)$.

The time complexity of TEST follows directly from Theorem 3.2. □

Our PTAS is presented next. As in [12], we will use LB to denote a lower bound on $c(\mathscr{T},\mathscr{W})$ and use UB to denote an *approximate* upper bound on $c(\mathscr{T},\mathscr{W})$ such that $2\times\mathsf{UB}$ is a valid upper bound on $c(\mathscr{T},\mathscr{W})$ and that $\mathsf{UB}\geqslant\mathsf{LB}$. Our PTAS starts with efficiently computable values of LB and UB and uses bisection to drive the ratio UB/LB to below 2. Once we have a pair of UB and LB such that $\mathsf{UB}/\mathsf{LB}\leqslant 2$, we can use $\theta=(n-1)/(\mathsf{LB}\times\varepsilon)$ to scale the edge cost and use $\zeta=2\times\mathsf{UB}\times\theta$ as an upper bound to compute an $(1+\varepsilon)$-approximation to the weight constrained Steiner tree.

Algorithm 3. PTAS for weight constrained minimum cost Steiner tree in an S–P graph.

```
Step_1  set LB and UB to their initial values such that 2×UB ⩽ LB×(n − 1);
Step_2  if UB ⩽ 2×LB then
          goto Step_3;
          else
          let 𝒞 := √(UB×LB);
          if TEST(𝒞,1) = NO, set LB = 𝒞;
          if TEST(𝒞,1) = YES, set UB = 𝒞;
          goto Step_2;
          endif
Step_3  set θ := (n − 1)/(LB×ε); ζ := θ×UB×2;
        set c_θ(e) := ⌊θ×c(e)⌋ for every e ∈ E;
        apply Algorithm 1 to compute a cost constrained minimum weight Steiner
        tree using the scaled cost function c_θ
```

The initial values of LB and UB can be computed as follows. Let $c_1<c_2<\cdots<c_k$ be the different edge cost values. Clearly we have $k\leqslant 2n-3$. Use G_j to denote the graph obtained by deleting all edges in G whose cost is more than c_j. Note that in $O(n)$ time, we can compute a minimum weight Steiner tree in G_j for $j=1,2,\ldots,k$. Let J be the smallest index j such that the weight of the minimum weight Steiner tree in G_j is no more than $\mathscr{W}$. Therefore, $c(\mathscr{T},\mathscr{W})$, the cost of an optimal solution to the

weight constrained Steiner tree must be in the interval $[c_J, c_J \times (n-1)]$. We may use c_J and $c_J \times (n-1)/2$ as the initial values for LB and UB, respectively. This can be done in $O(n \log n)$ time by bisection on the different edge cost values.

Theorem 4.2. *If there is no weight constrained Steiner tree spanning the target vertices in $\mathcal{T}$, we will find this out during our computation of the initial values of* UB *and* LB $(J == \infty)$. *If $\mathcal{T}$ has a weight constrained Steiner tree, Algorithm* 3 *finds a weight constrained Steiner tree T such that $w(T) \leqslant \mathcal{W}$ and $c(T) \leqslant (1+\varepsilon) \times c(\mathcal{T}, \mathcal{W})$. Furthermore, the time complexity of Algorithm* 3 *is* $O(n^3 \times (\log\log n + (1/\varepsilon^2)))$.

Proof. The claim follows from standard techniques [2,7,12,18] and Theorems 2–4. Let us use $\mathsf{LB}^{[0]}$ and $\mathsf{UB}^{[0]}$ to denote the initial lower bound and approximate upper bound for $c(\mathcal{T}, \mathcal{W})$ and use $\mathsf{LB}^{[k]}$ and $\mathsf{UB}^{[k]}$ to denote the lower bound and approximate upper bound for $c(\mathcal{T}, \mathcal{W})$ obtained after k iterations (executions of Step_2) for $k = 1, 2, 3, \ldots$. It follows from our analysis before the presentation of Algorithm 3 that

$$\log \frac{\mathsf{UB}^{[k]}}{\mathsf{LB}^{[k]}} \leqslant \frac{1}{2} \times \log \frac{\mathsf{UB}^{[k-1]}}{\mathsf{LB}^{[k-1]}} \leqslant \frac{1}{2^k} \times \log \frac{(n-1)}{2}. \tag{4.1}$$

As a result, we would have $\mathsf{UB}^{[k]}/\mathsf{LB}^{[k]} \leqslant 2$ after k iterations where k is no more than $\log\log n$ and log is the base-2 logarithm. Since $2 \times \mathsf{UB}$ is always a valid upper bound on $c(\mathcal{T}, \mathcal{W})$, it follows from Theorems 3.2 and 4.1 that Step_3 computes an $(1+\varepsilon)$-approximation to the weight constrained Steiner tree in $O(n \times (n/\varepsilon)^2)$ time. This completes the proof. □

5. Conclusions

In this paper, we have studied the WCSTP on a very important class of sparse graphs–S–P graphs. Although most Steiner tree problems are S–P graphs, it is shown that the weight constrained minimum cost Steiner tree problem on S–P graphs is NP-hard. On the positive side, we have presented a strongly polynomial time approximation scheme for this problem, which has applications to computer networks as well as computational biology. A more challenging problem is the WCSTP on a general graph.

Appendix A. NP-completeness of WCSTP in S–P graphs

In this section, we prove that the weight constrained shortest path problem [7] in S–P graphs is NP-hard. After we have arrived at this proof, we found that the proof by Wang and Crowcroft [15] also proves the same fact. This proof is included here as an aid to the referees.

Theorem A.1. *The weight constrained shortest path problem in S–P graphs is NP-hard.*

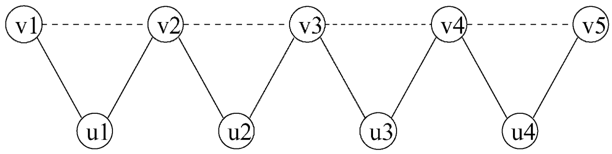

Fig. 1. Hardness proof of WCSPP.

Table 1
Initialization of arc measures

Initialization of measures
$st(\alpha,\xi)=w(e)$ if $c(e)\leqslant\xi$, BIG otherwise;
$dt(\alpha,\xi)=0$;
$yn(\alpha,\xi)=$ BIG if $y\in\mathcal{T}$, 0 otherwise;
$ny(\alpha,\xi)=$ BIG if $x\in\mathcal{T}$, 0 otherwise;
$nn(\alpha,\xi)=$ BIG if $x\in\mathcal{T}$ or $y\in\mathcal{T}$, 0 otherwise;
$none(\alpha,\xi)=$ BIG if $x\in\mathcal{T}$ or $y\in\mathcal{T}$, 0 otherwise.

Proof. Now we transform PARTITION to the weight constrained shortest path problem in S–P graphs.

Suppose we have a set S with n elements and every element a has a corresponding size $s(a)\in Z^+$, $\sum_{a\in S}s(a)=2A$. We construct a S–P graph (see Fig. 1) as follows.

- $V=\{v_1,v_2,\ldots,v_{n+1},u_1,u_2,\ldots,u_n\}$,
- $E=\{(v_i,v_{i+1})\,|\,i=1,2,\ldots,n\}\cup\{(v_i,u_i),(v_{i+1},u_i)\,|\,i=1,2,\ldots,n\}$.
- For $e=(v_i,v_{i+1})$, $c(e)=s(a_i), w(e)=0$.
- For $e=(v_i,u_i)$ or (v_{i+1},u_i), $c(e)=0, w(e)=\frac{1}{2}s(a_i)$.

For this S–P graph, we want to find a shortest path from v_1 to v_{n+1} such that the total weight of this path is no larger than A.

If we can find a partition of $S=S_1\cup S_2$ such that $\sum_{a\in S_1}s(a)=\sum_{a\in S_2}s(a)=A$, then we have a path from v_1 to v_{n+1} consist of the edges (v_i,v_{i+1}) for $a_i\in S_1$, and $(v_i,u_i),(u_i,v_{i+1})$ for $a_i\in S_2$. The total cost and total weight of this path are all equal to A. This path is obviously the shortest path under the restriction of total weight no larger than A.

Conversely, assume that we have find a shortest path from v_1 to v_{n+1} with the total weight no larger than A. From the construction of the S–P graph, we can get a partition of S by setting $S_1=\{a_i\,|\,(v_i,v_{i+1})$ is in the shortest path$\}$.

From above discussion, we know that the weight constrained shortest path problem in S–P graph is NP-hard. □

References

[1] S. Arora, Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems, J. ACM 45 (1998) 753–782.

[2] G.T. Chen, G.L. Xue, K-pair delay constrained minimum cost routing in undirected networks, SODA2001–ACM-SIAM Symp. on Discrete Algorithms, pp. 230–231.
[3] C.J. Colbourn, G. Xue, Grade of service Steiner trees on a series–parallel network, in: D.-Z. Du, J.M. Smith, J.H. Rubinstein (Eds.), Advances in Steiner Trees, Kluwer Academic Publishers, Dordrecht, 2000, pp. 163–174.
[4] D.Z. Du, F.K. Hwang, An approach for proving lower bounds: solution of Gilbert–Pollak conjecture on Steiner ratio, IEEE FOCS'90, pp. 76–85.
[5] A.M. Farley, Networks immune to isolated failures, Networks 11 (1981) 255–268.
[6] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H Freeman and Company, San Francisco, CA, 1979.
[7] R. Hassin, Approximation schemes for the restricted shortest path problem, Math. Oper. Res. 17 (1992) 36–42.
[8] F.K. Hwang, D.S. Richards, Pawel Winter, The Steiner tree problem, in: Annals of Discrete Mathematics, vol. 53, North-Holland, Amsterdam, 1992.
[9] O. Ibarra, C. Kim, Fast approximation algorithms for the knapsack and sum of subsets problems, J. ACM 22 463–468.
[10] X. Jia, L. Wang, Group multicast routing using multiple minimum Steiner trees, J. Comput. Commun. (1997) 750–758.
[11] T. Jiang, L. Wang, Computing shortest networks under a fixed topology, in: D.-Z. Du, J.M. Smith, J.H. Rubinstein (Eds.), Advances in Steiner Trees, Kluwer Academic Publishers, Dordrecht, 2000, pp. 39–62.
[12] D.H. Lorenz, D. Raz, A simple efficient approximation scheme for the restricted shortest path problem, Oper. Res. Lett. 28 (2001) 213–219.
[13] S. Sahni, General techniques for combinatorial approximations, Oper. Res. 35 (1977) 70–79.
[14] J.A. Wald, C.J. Colbourn, Steiner trees, partial 2-trees, and minimum IFI networks, Networks 13 (1983) 159–167.
[15] Z. Wang, J. Crowcroft, Quality-of-service routing for supporting multimedia applications, IEEE J. Selected Areas Commun. 14 (1996) 1228–1234.
[16] P. Winter, Generalized Steiner problem in series–parallel networks, J. Algorithms 7 (1986) 549–566.
[17] G.L. Xue, G.H. Lin, D.Z. Du, Grade of service Steiner minimum trees in the Euclidean plane, Algorithmica, in press.
[18] G.L. Xue, W. Xiao, An FPTAS for minimum cost delay constrained multicast tree under a Steiner topology, with applications, submitted for publication.